

Utility Patent Application
Z-44-1 US

What is claimed is:

1. A network server comprising shared memory and a cache mechanism for caching intermediate code files in said shared memory.
2. A network server according to claim 1, wherein said network server is a Web server.
3. A network server according to claim 1, wherein said cache mechanism comprises a shared memory replicator.
4. A network server according to claim 1, wherein said cache mechanism comprises a shared memory manager.
5. An intermediate code file according to claim 1, wherein said intermediate code is compiled from a PHP script.
6. A shared memory manager for caching intermediate code in shared memory.
7. A shared memory replicator for copying data structures into shared memory, wherein said data structure placed in said shared memory maintains the relationships between elements in said data structure in a way said data structure remains intact.
8. A process for caching intermediate code files in shared memory comprising the steps of:
 - receiving a script from an input port and sending said script to an scripting engine interface; determining whether an intermediate code product of said script is cached in said shared memory by checking an access-name registry compiling said script into an intermediate code;

and if said intermediate code does not exist in said cache script registry, copying said intermediate code from regular memory to shared memory; and saving pointers for each segment saved in said share memory.

5

9. A process for maintaining and updating intermediate code files in cached shared memory comprising the steps of:

- a. initializing a translation table for mapping old addresses to new address; and
- 10 b. copying said data structure from regular memory to shared memory; and
- c. checking if memory blocks in said data structure exists in said translation table;
- 15 d. allocating a block in shared memory if said memory block doesn't exist in said translation table;
- e. storing entries for mapping correlation of non-shared block memory addresses to a shared memory addresses of said allocated shared blocks in said translation table;
- f. updating pointers with new shared memory addresses using a translation function;
- 20 g. checking if every memory address has been transferred to said shared memory using said translation table;
- h. changing contents of pointer to said shared memory address if said pointer exists in said translation table;
- 25 i. updating said translation table.

10. The combination of a scripting engine and a shared-memory cache in which scripts from said scripting engine are cached.

30

Utility Patent Application
Z-44-1 US

11. The combination of claim 11, wherein said scripting engine is capable of servicing at least two processes concurrently.

5 12. The combination of claim 11, wherein said scripting engine is capable of re-caching a modified script.

13. The combination of claim 11, wherein the scripting engine resides in a server.

10 14. The combination of claim 11, wherein said scripting engine is a PHP scripting engine.

15 15. The combination of claim 11, wherein said scripting engine is a Perl scripting engine.

16. The combination of claim 11, wherein said scripting engine is a Python scripting engine.

17. The combination of claim 13, further comprising a network server.

20 18. The combination of claim 17, wherein said network server is a Web server.

25 19. The combination of claim 11, wherein the cache shared memory identifies that 2 labels refer to the same underlying file.

Utility Patent Application
Z-44-1 US

20. The combination of claim 11, wherein said scripting engine is capable of recognizing which files should not be cached.

21. The process according to claim 8, wherein said determining step
5 comprises comparing the timestamp of said received script to the timestamp of a previously cached version of said script.

22. The process according to claim 8, wherein said received script is re-
cached if the timestamp of said received script is different from the
10 timestamp of said previously cached version of said script.

23. The process according to claim 13, further comprising referring to a full
path registry to determine whether to recache a previously cached script.

24. The process according to claim 17, further comprising retrieving said
15 cached script, using said full path according to an entry of said full path registry.

25. A shared memory replicator for copying data structure into shared
20 memory, wherein said data structure maintains the original relationships between elements in said data structure such that the data structure remains intact and usable.